OWASP TOP 10

Hai Hoang

NASH TECH
The Power to Innovate

Agenda

- WHAT
- WHY
- WHERE
- TOP 10
- QA

NASH TECH
The Power to Innovate

WHAT

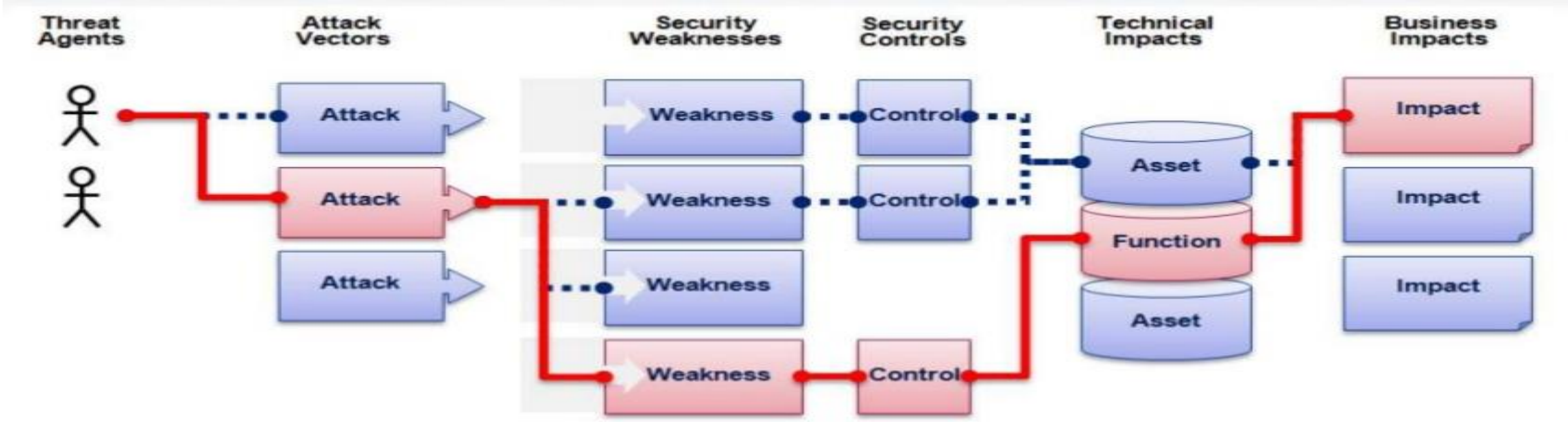
Introduction

- **O**pen **W**eb **A**pplication **S**ecurity **P**roject
  - A list of the 10 most critical web application security risks
  - Open community & non-profit
  - **Risk based** approach
- Core purpose
  - Safety
  - Security

NASH TECH
The Power to Innovate

WHY

OWASP TOP 10: Risk Rating Methodology



Threat Agent	Attack Vector	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact
?	1 Easy	Widespread	Easy	Severe	?
	2 Average	Common	Average	Moderate	
	3 Difficult	Uncommon	Difficult	Minor	

OWASP TOP 10: Risk Rating Methodology (cont.)

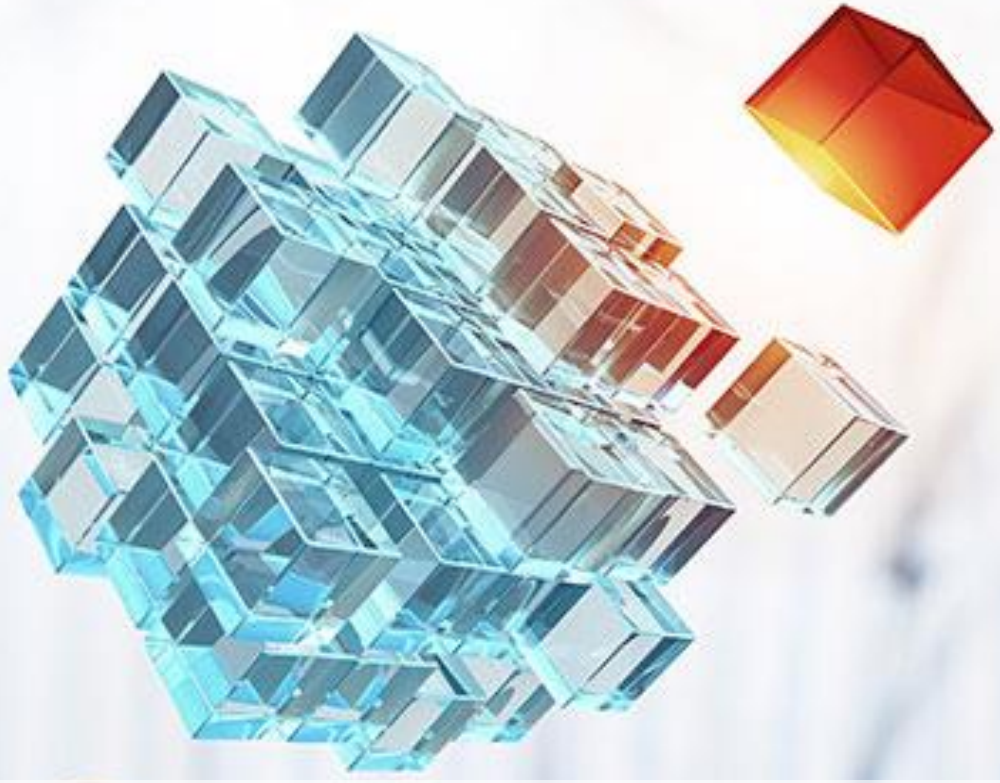
Weighted Risk Rating = Probability * Impact

Threat Agent	Attack Vector	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact
?	1 Easy	Widespread	Easy	1 Severe	?
	Average	2 Common	2 Average	Moderate	
	Difficult	Uncommon	Difficult	Minor	

$(1+2+2)/3 = \mathbf{1.66}$

$1.66 * 1 = \mathbf{1.66}$

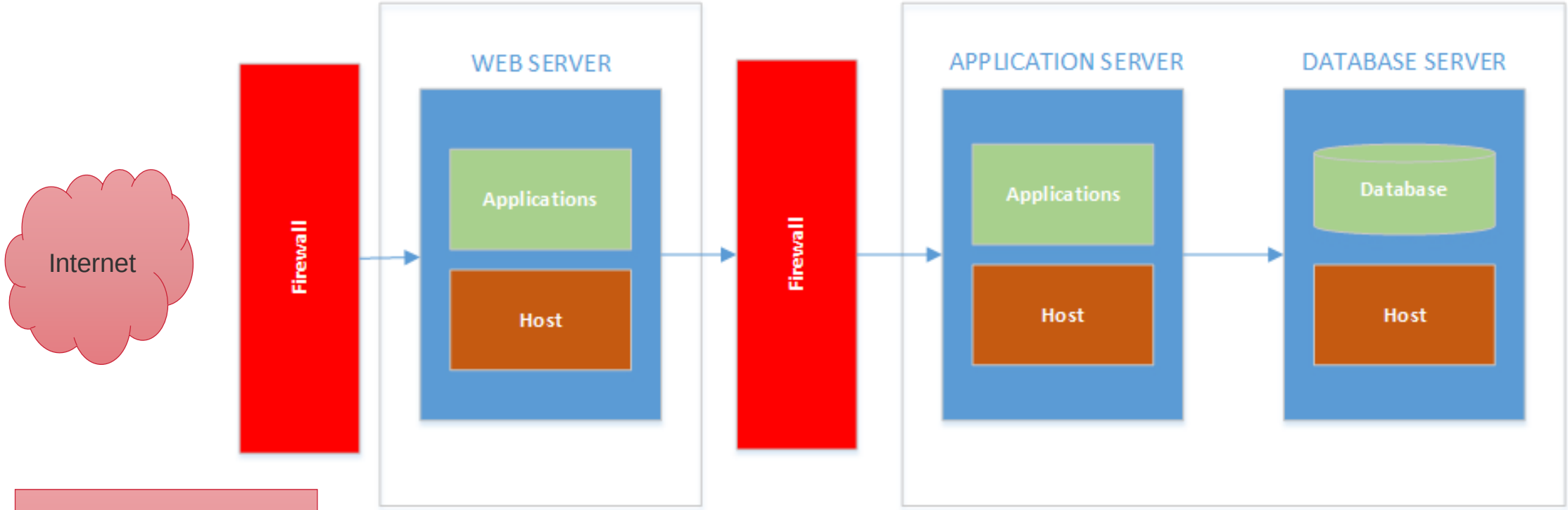
NASH TECH
The Power to Innovate



WHERE

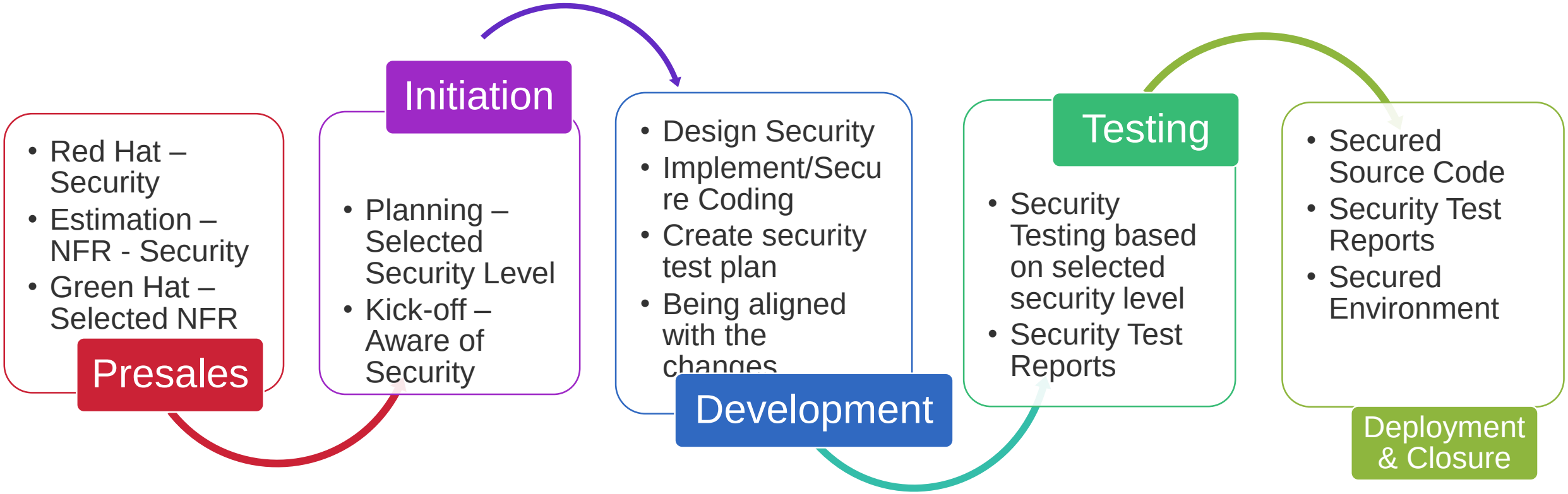
- Input Validation
- Authentication
- Authorization
- Configuration Management
- Sensitive Data
- Session Management
- Cryptography
- Parameter Manipulation
- Exception Management
- Auditing and Logging

OWASP
Open Web Application Security Project



- Router
- Firewall
- Switch

- Patches and Updates
- Services
- Protocols
- Accounts
- Files and Directories
- Shares
- Ports
- Registry
- Auditing and Logging
- Anti-virus software (AV)
- System Hardening
- Forefront Identity Manager (FIM)


Presales
• Red Hat – Security
• Estimation – NFR - Security
• Green Hat – Selected NFR
Initiation
• Planning – Selected Security Level
• Kick-off – Aware of Security
Development
• Design Security
• Implement/Secure Coding
• Create security test plan
• Being aligned with the changes
Testing
• Security Testing based on selected security level
• Security Test Reports
Deployment & Closure
• Secured Source Code
• Security Test Reports
• Secured Environment

Terms in general

Stateless - Stateful

- Stateless -> Stateful
  - URL Rewriter
  - HTML Form
  - **Session**
  - **Cookie**

Session

- Created on server
- Server
- Attribute:
  - Value
  - Expiration

Cookie

- Created on server
- Browser
- Attribute:
  - Domain
  - Path
  - HttpOnly
  - Secure

Http StatusCode

- 1xx – Information: 100 (Continue), 102 (Processing)
- 2xx - Success: 200 (OK), 202 (Accepted)
- 3xx - Redirection: 302 (Found), 304 (Not Modified)
- 4xx - Client Error: 403 (Forbidden), 404 (Not Found)
- 5xx - Server Error: 500 (Internal Server Error), 503 (Service Unavailable)

OWASP TOP 10

NASH TECH
The Power to Innovate

OWASP TOP 10

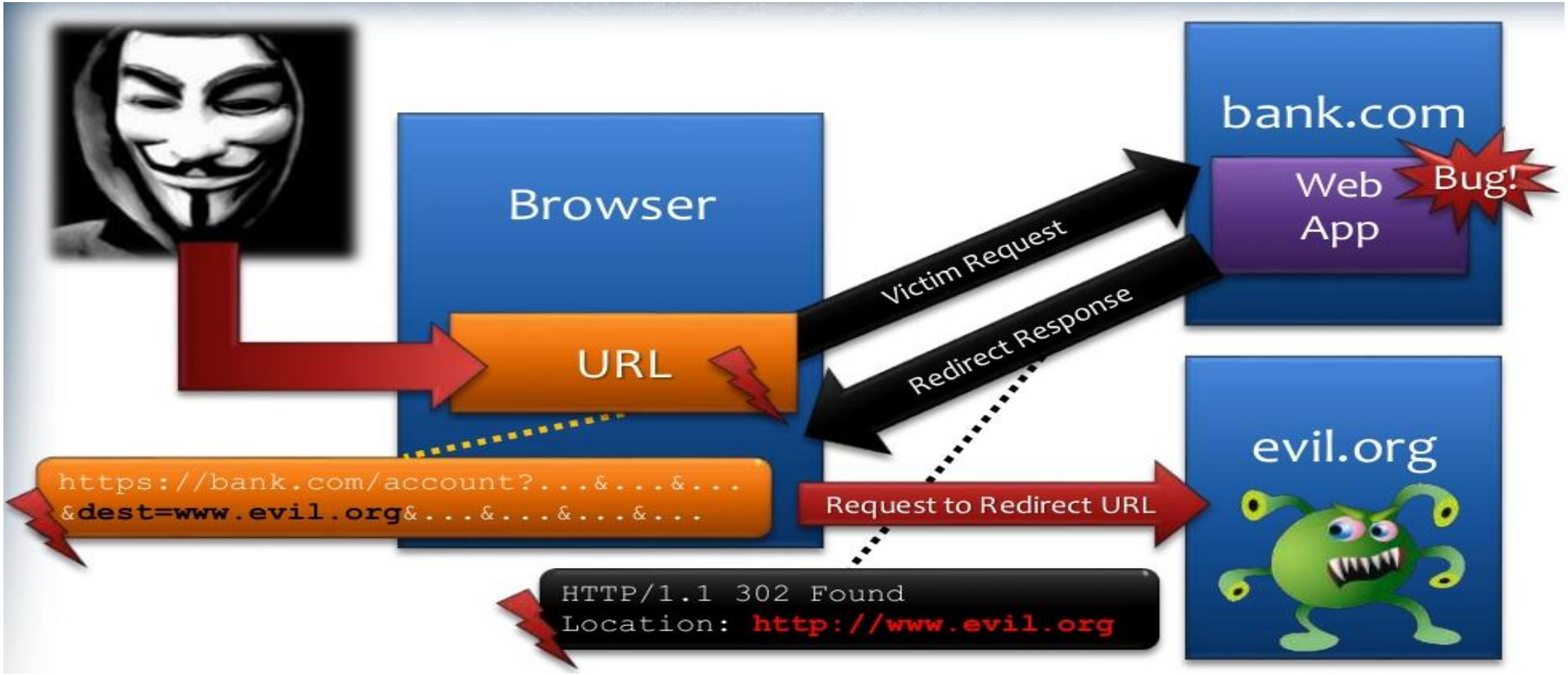
- A10: Un-validated Redirects and Forwards
- A9: Using Known Vulnerable Components
- A8: Cross-Site Request Forgery (CSRF)
- A7: Missing Function Level Action Control
- A6: Sensitive Data Exposure
- A5: Security Misconfiguration
- A4: Insecure Direct Object Reference
- A3: Cross-Site Scripting (XSS)
- A2: Broken Authentication & Session Management
- A1: Injection
- Ex-A6/2007: Information Leakage and Improper Error Handling

OWASP TOP 10

- **A10: Un-validated Redirects and Forwards**

A10: Risks

- Redirect & Forward



A10: Impact

- **Redirected** to **phishing/malware site.**
- **Bypass security checks**, allowing **unauthorized function** or **data access**.

A10: Prevent

- **Avoid** using **redirects and forwards**.
- **Don’t involve user parameters**
- If you ‘**must**’ **involve user parameters**, the either
  - Use **server side mapping**
  - **Validate** each **parameter**
- Use a **secure Redirect API** (ESAPI)

OWASP TOP 10

- A10: Un-validated Redirects and Forwards
- **A9: Using Known Vulnerable Components**

A9: Risks

- Developed by third-party
- W**eaknesses**
  - Injection, broken access control, XSS, etc.

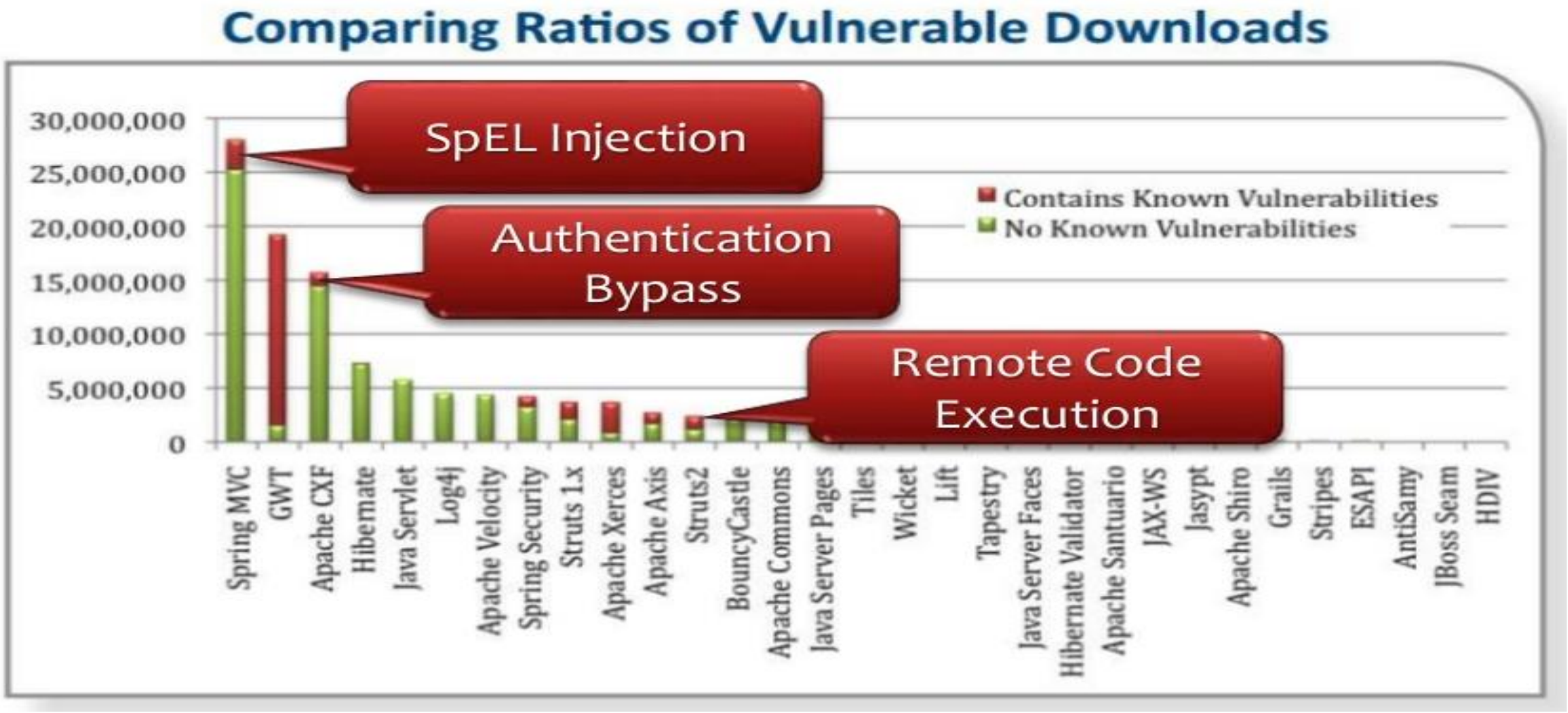
A9: Impact

- Host takeover and data compromise.

A9: Prevent

- **Identify components and its version.**
- **Monitor & keep up to date.**
- **Restrict for using unapproved components.**

A9: Prominent Library Vulnerabilities



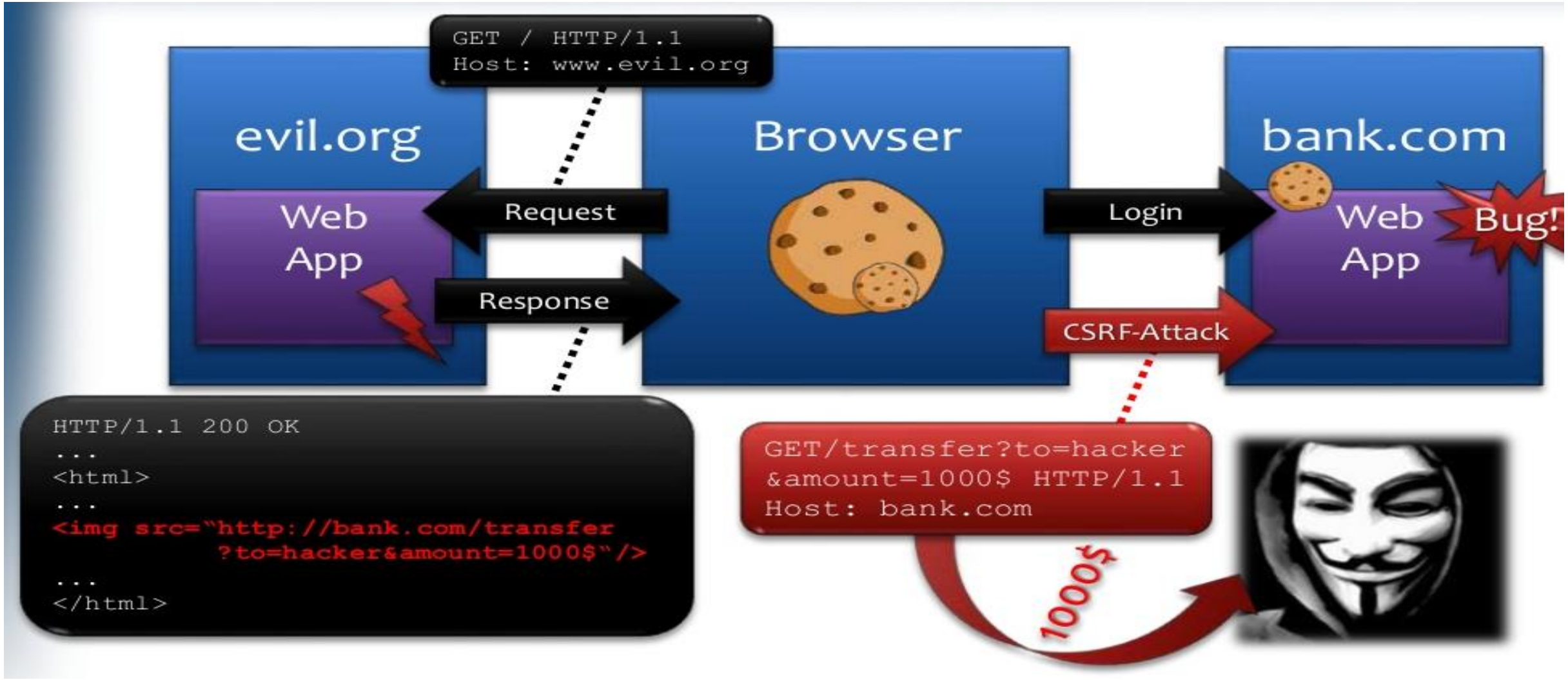
OWASP TOP 10

- A10: Un-validated Redirects and Forwards
- A9: Using Known Vulnerable Components
- **A8: Cross-Site Request Forgery (CSRF)**

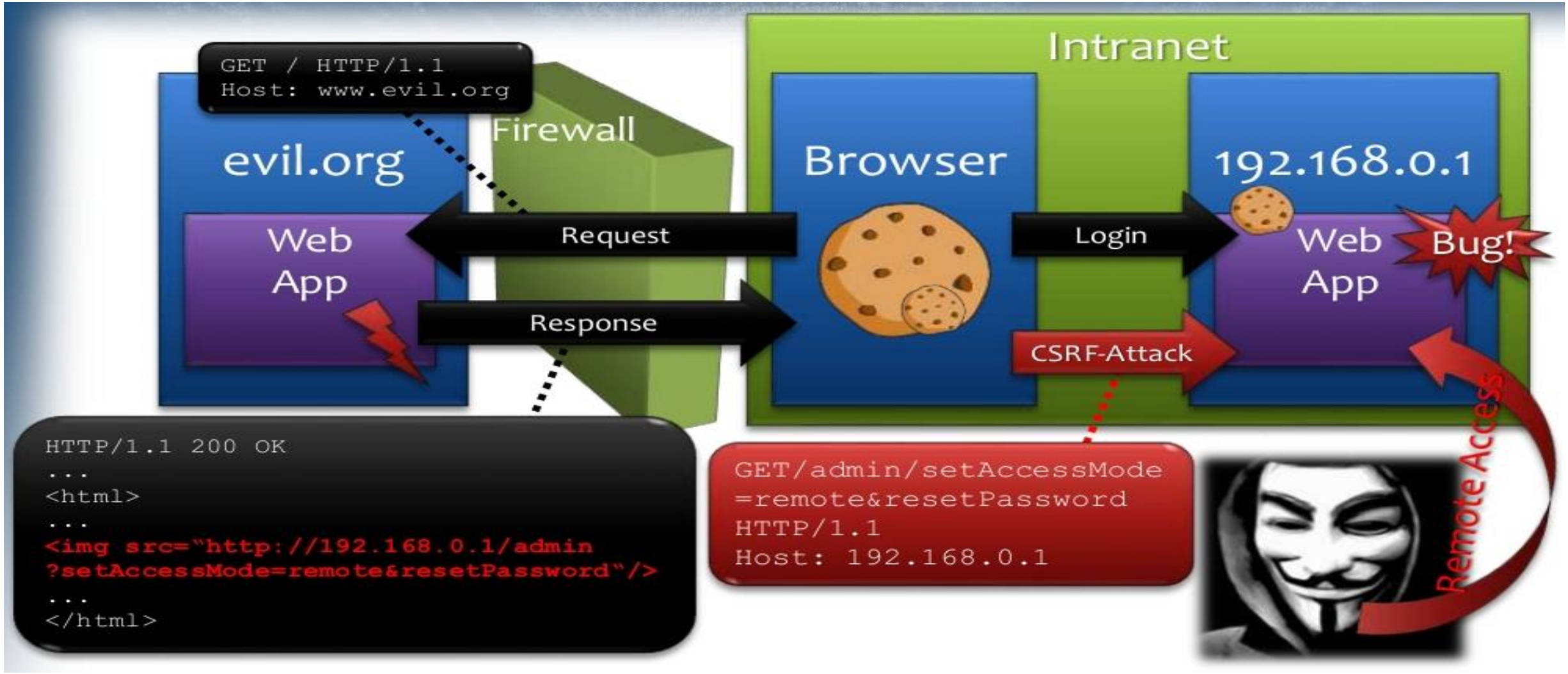
A8: Risks

- **Command** to **a vulnerable web application.**
- **Including user authentication data** with **each request.**

A8: Scenarios



A8: Scenarios (cont.)



A8: Impact (*)

- **Initiate transactions**
  - Transfer funds, logout user, close account,…
- **Access sensitive data**
- **Elevation of privilege** (change account details…)
- **Denial of Service**
- **Spoofing and tampering**

A8: Prevent

- **Using secret token** for **all sensitive requests.**

OWASP TOP 10

- A10: Un-validated Redirects and Forwards
- A9: Using Known Vulnerable Components
- A8: Cross-Site Request Forgery (CSRF)
- **A7: Missing Function Level Action Control**

A7: Risks

- **Common mistakes**
  - **Displaying** only **authorized links** and **menu choices.**
  - **Forges** direct **access** to **`unauthorized` pages.**

A7: Impact

- I**nvoke functions** and **services** they’re **not authorized.**
- **Access** other **user’s account** and **data.**
- **Perform privileged actions** (Elevation of privilege)**.**

A7: Prevent

- **Restrict access** to **authenticated users** (if not public).
- **Enforce** using **RBAC** / **CBAC** / **ACL** technical.
- **Disallow request** to **unauthorized page** types (config files, log files, source files…).

OWASP TOP 10

- A10: Un-validated Redirects and Forwards
- A9: Using Known Vulnerable Components
- A8: Cross-Site Request Forgery (CSRF)
- A7: Missing Function Level Action Control
- **A6: Sensitive Data Exposure**

A6: Risks

- Storing sensitive data
  - What?
  - Where?
  - How?

A6: Risks (cont.)

- Sensitive Data is **not encrypted.**
- Using **self-made crypto algorithms.**
- **Store Keys and Passwords** in **Source Code.**
- **Store Keys/Certificates** in **unsafe location**.
- Continued **usage** of **weak crypto algorithms**
  - MD5, SHA-1, RC3, RC4

A6: Impact

- A**ccess** or **modify confidential/ private information**.
- E**xtract secrets** to **use in additional attacks**.
- **Company embarrassment**, **customer dissatisfaction**, and **loss of trust.**

A6: Prevent

- **Verify your architecture**
  - **Identify** all **sensitive data.**
  - **Identify all the places that data is stored.**
  - Use encryption to **counter the threat**, **don’t just `encrypt`** the **data**.
- **Protect with appropriate mechanisms**
  - **File** encryption, **database** encryption, **data element** encryption.
  - **Use TLS** on all **connections** with **sensitive data.**

A6: Prevent (cont.)

- **Use the mechanisms** correctly
  - Use **standard strong algorithms** (AES, RSA, SHA-256).
  - **Generate**, **distribute**, and **protect keys** properly.
  - Be **prepared** for **key change**.
- **Verify the implementation**
- Be specially **careful in unknown Networks** (WLAN Hotspots, Internet Café…).

NASH TECH
The Power to Innovate

Q&A

THANK YOU

www.nashtechglobal.com